

I hereby certify that this paper and/or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 CFR §1.10 on the date indicated below and is addressed to: Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Michelle Fitzgummings
Signature

DATE OF DEPOSIT: June 30, 2003

EXPRESS MAIL LABEL NO.: EV331728273US

Inventor(s): Marcus A. Baker, Jeffrey B. Williams, and Sheldon J. Sigrist

METHOD AND SYSTEM FOR SAVING THE STATE OF INTEGRATED CIRCUITS UPON FAILURE

FIELD OF THE INVENTION

The present invention relates to the operation of integrated circuits, and more particularly to the recovery from failures of processors and other integrated circuits.

5 BACKGROUND OF THE INVENTION

Interated circuits are used in a wide variety of applications. For example, servers are used to provide a variety of processing and storage functions in computer systems. Application Specific Integrated Circuits (ASICs), microprocessor chips, memory chips, and other integrated circuit chips of various forms provide much of the control and functionality of the servers. At various times,
10 however, integrated circuits and chipsets can fail during operation due to any of many possible reasons.

In some servers and other computer devices, one or more microprocessors or other circuits can be used to monitor the operation of the server (or other device). This "service processor" can monitor the operation of the server and determine if and when a failure occurs in the server, and
15 whether that failure is caused by the hardware or software of the server itself, or by some other

cause.

When a failure does occur in the server system, the service processor can check if it is an uncorrectable error, i.e., an error which requires servicing by the server operator or provider, and which cannot be easily or immediately be corrected by the service processor. If the error is uncorrectable, each ASIC, microprocessor, or other integrated circuit processing chip of the server can lock the state of all of its internal latches at the time of the error. The service processor can then dump or save the data contents of certain registers of the locked chips, so that the data in these registers can be later examined to help determine the nature of the error and the source or cause of the error. The service processor dumps this register data contents to the service processor's memory.

Some problems can occur with the above situation where the service processor dumps an integrated circuit chip's registers after an uncorrectable error. One problem is that the service processor has a finite memory capacity used mainly for its own functions, and thus cannot dump and save the contents of even a substantial number of the latches of the integrated circuit chip. Another problem is that the service processor is programmed beforehand to dump specific registers of the integrated circuit chip. The registers chosen to be dumped by the service processor are usually determined and selected based on previous debugging sessions of the integrated circuit chip, e.g., particular errors were found in earlier debugging stages and the pertinent registers for those found errors are chosen to be dumped for future chip failures. Therefore, there is a chance that when a new type of failure occurs in the integrated circuit chip, not all of the correct latches and registers of the chip that will lead to a successful determination of the error will have been dumped by the service processor.

SUMMARY OF THE INVENTION

The invention of the present application provides a method and system for saving the state of integrated circuits upon failure. In a first aspect of the present invention, a system for saving the state of an integrated circuit includes a non-volatile memory, and a state-saving controller coupled to the non-volatile memory and coupled to the integrated circuit, where the state-saving controller saves the state of the integrated circuit to the non-volatile memory when a failure occurs in the integrated circuit. In a second aspect of the present invention, a method for saving the state of an integrated circuit includes determining that an error has occurred in the operation of the integrated circuit, and saving the state of the integrated circuit to a non-volatile memory coupled to the integrated circuit, the state saved after the error has been detected. In a third aspect of the present invention, a computer readable medium includes program instructions to be implemented by a computer, the program instructions implementing steps for saving the state of an integrated circuit, the steps including determining that an error has occurred in the operation of the integrated circuit, and saving the state of the integrated circuit to a non-volatile memory coupled to the integrated circuit, the state saved after the error has been detected.

The present invention provides methods and apparatus for saving the state of an integrated circuit when a failure occurs in the circuit. Since the state of the integrated circuit is saved to a dedicated memory, substantially all the latches of the integrated circuit can be saved instead of only particular registers, thus allowing a thorough examination of the state of the integrated circuit at the time of failure and leading to potentially more effective debugging operations and fixes of the circuit. These and other advantages are set forth in greater detail in the specification and drawings provided below.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system of the present invention for saving the state of an integrated circuit;

Figure 2 is a block diagram of a second embodiment of the present invention for saving the state of an integrated circuit;

Figure 3 is a block diagram of a third embodiment of the present invention for saving the state of an integrated circuit; and

Figure 4 is a flow diagram illustrating a method of the present invention for saving the state of an integrated circuit.

DETAILED DESCRIPTION

The present invention relates to the operation of integrated circuits, and more particularly to the recovery from failures of processor and other integrated circuits. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

FIGURE 1 is a block diagram of a system 10 of the present invention. System 10 includes an Application Specific Integrated Circuit (ASIC) 12, a bus master device 14, a bus 16, and a memory device 18.

ASIC 12 is included in a server or other type of computer device. The function of the ASIC

can vary greatly depending on the specific application for which it is provided and programmed. For example, in a server system, the ASIC 12 can be used in the general operation of the server, e.g., to perform computations and other functions, store and retrieve data from memory, communicate with other components of the server system, etc. Or the ASIC 12 can be used in more specific and narrow applications. The server or other computer device with which the ASIC 12 operates can include various other well-known components, including microprocessor(s), memory, storage devices, communication and I/O devices, etc. For example, in one embodiment, ASIC 12 can be one or more chips in the Pecos chipset from IBM Corp. or similar types of chips.

In other embodiments, other types of integrated circuits 12, such as microprocessors, other processors, and the like, can be used instead of ASIC 12 in all the embodiments of the invention. For example, integrated circuit chips such as Field Programmable Gate Arrays (FPGAs) or Custom Programmable Logic Devices (CPLDs), can be used in any embodiment of the present invention.

ASIC 12 has a number of latches for storing data for and during various uses and operations of the ASIC. For example, some latches are used as registers for the ASIC, e.g., for the temporary storage of data during operations and indications of the states of components of the ASIC. Other latches may save the state of data paths of the ASIC, the state of memory of the ASIC, or another relevant state. These types of latches may be useful for determining the state of the ASIC upon failure.

Bus master device 14 is included in system 10 and operates as a state-saving controller for the present invention to manage the retrieval and storage of data from the ASIC 12 when an uncorrectable error occurs in the system. Bus master 14 also can control the bus 16, e.g., control what devices can send and receive data over the bus 16. For example, bus master 14 can be a controller that is an integrated circuit, an ASIC, a microprocessor, or other circuit that can be

programmed via software and/or hardware. In the present embodiment of Fig. 1, bus master 14 is implemented as an external, separate unit or chip from the ASIC 12. In the present invention, bus master 14 and its state-saving controller are dedicated to saving the state of the ASIC to memory, and do not perform any other functions or act as a general service processor (e.g., a service
5 processor typically provides many functions including monitoring temperatures of server components, allowing remote control of the server or computer, powering the server/computer on or off, etc.).

Bus 16 is coupled to the ASIC 12 and to the bus master 14 and is used to transfer data from the ASIC to a memory 18 (described below). In the described embodiment, bus 16 is a serial bus,
10 but the bus can be a different type of bus in other embodiments (parallel bus, etc.) The bus master 14 can communicate with the ASIC 12 over the serial bus 16 without having to first communicate with a CPU or other microprocessor of the computer system. The serial bus 16 allows serial transfer of data to or from the ASIC to other devices connected to the serial bus. In the preferred embodiment, the bus master 14 and the memory 18 are the only two devices on the serial bus 16
15 besides the ASIC 12. Serial bus 16 can be any of a variety of types of buses. For example, in one embodiment, bus 16 is an I2C bus (Inter-IC bus) and bus master 14 is an I2C master. The I2C bus is a multi-master bus that can be used to allow many types of integrated circuits and devices to communicate.

Non-Volatile Random Access Memory (NVRAM) 18 is included in system 10 and is in
20 communication with the ASIC 12 and the bus master 14 via serial bus 16. NVRAM 18 is a non-volatile memory and can hold data even when the power to the ASIC 12 and to the memory is discontinued, and thus is suitable to hold diagnostic data after system failures. Such data can be retrieved from the NVRAM at a time after failure for debugging purposes. For example, NVRAM

18 can be an Electrically Erasable Programmable Read Only Memory (EEPROM), static random access memory (SRAM), or other type of non-volatile memory. In alternate embodiments, other types of memory can be used which is able to reliably retain data after failures of the ASIC 12 or other components of the larger system of which ASIC 12 is a part.

5 In the present invention, bus master 14 reads the latches of ASIC 12 and stores the latch data in NVRAM 18 over the serial bus 16, i.e., the state of the ASIC 12 (between resets and power cycles of the server or computer device) is “dumped” to the NVRAM 18. In the described embodiment, NVRAM 18 thus acts as a slave serial device on the serial bus 16. The process of the present invention of saving the state of the integrated circuit 12 is described in greater detail with
10 respect to Fig. 4, below.

FIGURE 2 is a block diagram of a different embodiment 50 of the system of the present invention for saving the state of an integrated circuit. System 50 includes an ASIC 52, a bus master device 54, a bus 56, and NVRAM 58.

Integrated circuit 52, such as ASIC 52, performs tasks in the server or other computer
15 device according to its programming, similar to the ASIC 12 described with respect to Fig. 1. Other types of integrated circuits may also be used. Bus master 54 is connected to a serial bus 56 (or other type of bus) and can provide data from the latches of the ASIC to the serial bus and control the serial bus, similar to the bus master 14 described with respect to Fig. 1. However, the bus master 54 is internal to, i.e., embedded within, the ASIC 52 rather than being implemented as a
20 separate integrated circuit in a separate package. The embedding of the bus master 54 in ASIC 52 may allow the costs for producing these components to be reduced, but may require that the ASIC 52 be made more custom and specific to include this embodiment of the present invention.

Serial bus 56 and NVRAM 58 function similarly to their equivalent components in the

embodiment 10 of Fig. 1, where the NVRAM 58 can receive data from latches of the ASIC 52 over the serial bus 56 as directed by the bus master 54 upon failure of the ASIC 52, and the NVRAM can store that data in a non-volatile manner to allow the data to be later accessed for debugging purposes.

5 FIGURE 3 is a block diagram of a different embodiment 100 of the system of the present invention for saving the state of an integrated circuit. System 100 includes an ASIC and a serial bus 106.

 ASIC 102 is an integrated circuit that performs tasks in the server or other computer device according to its programming, similar to the ASICs 12 and 52 described above. Other types of
10 integrated circuits may also be used. In the embodiment of Fig. 3, a state-saving controller 104 and NVRAM 108 are internal to the ASIC, i.e., integrated and embedded with the ASIC 102 and in the same package.

 Since the NVRAM 108 is internal to the ASIC 102, no separate bus master device or similar external controller is needed to read the latches of the ASIC and send the latched data to the
15 NVRAM over a bus. Rather, the state-saving controller 104 functionality is included within the ASIC circuitry and programming (which was included in the bus master of the previous embodiments), so that the state-saving controller 104 can store the latch data to the internal NVRAM upon an uncorrectable failure of the ASIC 102.

 The integration of NVRAM 108 in the ASIC 102 may allow even greater consolidation of
20 the components of the present invention into a single package, reducing the space required on a circuit board and reducing the assembly time of the circuits, but potentially raising the costs of the circuit and requiring a more custom and specific ASIC. Also, an internal NVRAM 108 does not allow as easy access to the NVRAM memory to read its contents for debugging purposes, e.g., in

the embodiments of Figs. 1 and 2 having a separate NVRAM, the NVRAM chip can be separately removed from the circuit board and its contents read more easily. Furthermore, there may be a chance that the failure in the ASIC affects the state-saving controller functionality of the ASIC as well, so that the operation of the invention is compromised and no latches get saved to the NVRAM. Thus, it is desirable that the state-saving controller of the present invention be made as independent as possible from the remaining functionality of the ASIC 102.

A serial bus 106 (or other type of bus) is preferably included in the embodiment 100 to allow an appropriate testing device to be later connected to the bus 106 and the ASIC 102 so that the data in the NVRAM 108 can be downloaded to a testing device for diagnostic purposes.

In yet another embodiment, the bus master/state-saving controller 104 and the NVRAM 108 can be integrated in one integrated circuit chip that is separate from the ASIC 102 and is connected to the ASIC by a bus 106, such as a serial bus. The bus master/controller 104 would control the dumping of latches to the NVRAM similarly to the embodiments of Figs. 1 or 2.

FIGURE 4 is a flow diagram illustrating a method 200 of the present invention for saving the state of an integrated circuit upon failure. Programming instructions, data, code (or the equivalent in hardware such as logic gates, etc.) that implement the steps of this method can be stored by one or more computer readable media, such as memory, hard disk, magnetic tape, CD-ROM, DVD-ROM, storage device, or other medium, which is accessible to the appropriate controllers and devices described herein.

Any of the embodiments of Figs. 1-3 can use the presently described process; the embodiment of Fig. 1 is used as an example in this description. The process begins at 202, and in step 204, the process checks whether an unrecoverable error has occurred in the particular ASIC 12 which is being monitored. Errors, such as a timeout occurring on a bus, a system memory failing,

etc. can be detected using various well-known techniques. The error might be correctable, or unrecoverable. For example, a recoverable error can be a single bit error in data transferred over a parallel bus, which can be corrected by the ASIC using error checking and correction (ECC) information that is sent with the data. If, however (as an example), two or more bits are in error on the bus, the error may not be correctable with the ECC information and an unrecoverable error will have occurred. If an unrecoverable error has not occurred, the process continues to step 206, where the error is corrected or other appropriate action is taken. The process then returns to step 204 to check for an unrecoverable error during normal operation of the ASIC.

If an unrecoverable error has been detected in step 204, then in step 208, the process checks whether to dump the latches of the ASIC 12 based on the error. For example, the functionality of the present invention to dump the latches can be enabled or disabled by the user or operator or by programs running on the server, based on the desired operation of the ASIC. It should be noted that in the preferred embodiment, the process automatically dumps the state of the ASIC to memory (if enabled), rather than having to wait for a request from an external processor or other source to dump specific registers. In an alternate embodiment, the process can send out a notification to a user or operator or external processor (such as a service processor) that the error has occurred and wait for a specific request from the user/operator or external processor to dump the latches.

If latches are not to be dumped, the process continues to step 218, detailed below. If the latches are to be dumped, according to the present invention, then appropriate actions due to failure are taken, e.g., all internal clocks of the ASIC are stopped, etc., (the clocks can be stopped upon failure, regardless of whether latches are to be dumped or not in step 208) and the process continues to step 210.

In step 210, the non-volatile storage interface is initialized. The non-volatile storage

interface allows data to be written to the NVRAM 18; for example, a write protect bit of the NVRAM can be changed to allow data to be written to it, data in the NVRAM or other bits can be zeroed, etc. In next step 212, the contents of the current latch are saved (or “dumped”) to the NVRAM over the serial bus 16. The “current” latch is the first latch of the ASIC in the first iteration of this step, or another later-examined latch in later iterations. It should be noted that the current latch can be a latch that saves the state of registers of the ASIC, that saves the state of data paths of the ASIC, that saves the state of memory of the ASIC, or saves some other relevant state. Any or, in some embodiments, all of these types of latches may be useful for determining the state of the ASIC upon failure.

In step 214, the process checks whether all latches of the ASIC have been dumped to the NVRAM. If not, the process continues to step 216, where the next latch of the ASIC is examined, and the process then returns to step 212 to save the contents of that latch. Once all the latches have been dumped, the process continues from step 214 to step 218, in which an error flag is asserted. The error flag is a signal or other indication that is output from the ASIC to an appropriate device connected to the ASIC, where once the error flag is received, the appropriate device can take the appropriate action in the system and/or alert a user or operator of the computer device that includes the ASIC 12. For example, a separate service processor in a server can receive the error flag and can then reboot the server or remove power from the server, provide a notification to the user or operator of the server, etc. The process is then complete at 220.

A user or operator can then connect an external computer or other electronic device or debugging tool to the NVRAM in a debugging or test environment to read the data stored therein that was read from the latches of the ASIC. In some embodiments, the NVRAM can be removed from the circuit board to which it is connected so that its data can be read more conveniently, e.g., if

the circuit board cannot be powered. The diagnosis and remedy for the failure of the ASIC can be assisted by the data obtained from the NVRAM describing the state of the ASIC at the time of failure. This can be convenient when, for example, replacing nonfunctioning circuit boards in a server or computer device and then taking the replaced, nonfunctioning circuit board to the testing environment to learn more particularly about the failure.

In the preferred embodiment described above, all or substantially all the latches of the ASIC are dumped to the NVRAM. It should be noted, however, that in some other embodiments, not all of the latches of the ASIC need be dumped or saved. For example, there may be one or more (generally a small number) of latches of an ASIC that offer irrelevant, redundant, or insignificant information for determining the state of the ASIC for debugging purposes, and thus those latches need not be saved upon failure. In most embodiments, substantially all the latches of the ASIC are saved for simplicity and to preserve the state of the ASIC in NVRAM.

The present invention can also be used in conjunction with multiple ASICs or other types of integrated circuits. Multiple integrated circuits in a system can be monitored and their states saved according to the present invention. In one such embodiment, each integrated circuit has its own dedicated state-saving controller functionality, its own dedicated NVRAM, and its own process 200 that is independent of other ASICs and integrated circuits. If one ASIC fails, the process 200 is implemented to save the state of that ASIC, while the other ASICs may continue to run normally (unless they depend on data or signals from the failed ASIC, at which point they might also fail and run process 200, e.g., due to timeouts). In another embodiment, multiple ASICs or other integrated circuits are all connected to a single state-saving controller (such as a bus master) via a bus such as a serial bus, and are all connected to a single NVRAM which saves the states of all the ASICs connected thereto upon failure of one or more of the ASICs.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing
5 from the spirit and scope of the appended claims.